POSTLAB 5

1. Differentiate K-means and K-medoide algorithms with one example

Ans: -

Both k-means and k-medioids are used to produce clusters for which the objective that is meant to be minimized is the sum of the sum of squared distance of the points in some cluster to some other point over all clusters, or:argminS∑ki=1∑x∈Si||x−pi||22argminS∑i=1k∑x∈Si||x−pi||22

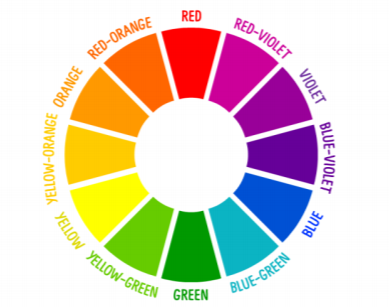
where S1,S2,...,SkS1,S2,...,Sk is a partitioning of the dataset to create clusters and pipiis some representative point for each cluster used to calculate the dissimilarity of a cluster.

For k-means, this point pipi is the mean of all the points within a cluster (μiμi),

so the objective in this case is ∑ki=1∑x∈Si||x−μi||22∑i=1k∑x∈Si||x−μi||22

For k-medioids, the representative point of each cluster is a datapoint inside that cluster, rather than the mean, so instead of μiμi, you would have some xi∈Si.xi∈Si. Each representative xixi for each cluster would be chosen in a way such that ∑x∈Si||x−xi||22∑x∈Si||x−xi||22 is minimal.

K-means is quite common in practice, but there are times when k-medioids is more useful. Sometimes, the mean of data is not meaningful (pun not intended). An example is hue colors.



If we were to train this with K-means on data with three dimensions (one for red, one for green, and one for blue) — and let’s say that during the training of the algorithm that we had a cluster with 50 points representing green pixel values and 10 points representing red pixel values — it would intuitively be much more meaningful to assign the representative point of that cluster to the red points than to their mean (yellow?)

1. Explain DBScan

Ans: -

Clustering analysis or simply Clustering is basically an Unsupervised learning method that divides the data points into a number of specific batches or groups, such that the data points in the same groups have similar properties and data points in different groups have different properties in some sense. It comprises of many different methods based on different evolution.E.g. K-Means (distance between points), Affinity propagation (graph distance), Mean-shift (distance between points), DBSCAN (distance between nearest points), Gaussian mixtures (Mahalanobis distance to centers), Spectral clustering (graph distance) etc.

Fundamentally, all clustering methods use the same approach i.e. first we calculate similarities and then we use it to cluster the data points into groups or batches. Here we will focus on **Density-based spatial clustering of applications with noise** (DBSCAN) clustering method.Clusters are dense regions in the data space, separated by regions of the lower density of points. The **DBSCAN algorithm** is based on this intuitive notion of “clusters” and “noise”. The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.

* **Why DBSCAN ?**  
  Partitioning methods (K-means, PAM clustering) and hierarchical clustering work for finding spherical-shaped clusters or convex clusters. In other words, they are suitable only for compact and well-separated clusters. Moreover, they are also severely affected by the presence of noise and outliers in the data.
* Real life data may contain irregularities, like –  
  **i)** Clusters can be of arbitrary shape.  
  **ii)**Data may contain noise.
* DBSCAN algorithm requires two parameters –

1. eps : It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to ‘eps’ then they are considered as neighbors. If the eps value is chosen too small then large part of the data will be considered as outliers. If it is chosen very large then the clusters will merge and majority of the data points will be in the same clusters. One way to find the eps value is based on the k-distance graph.
2. MinPts: Minimum number of neighbors (data points) within eps radius. Larger the dataset, the larger value of MinPts must be chosen. As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1. The minimum value of MinPts must be chosen at least 3.

In this algorithm, we have 3 types of data points.

Core Point: A point is a core point if it has more than MinPts points within eps.  
Border Point: A point which has fewer than MinPts within eps but it is in the neighborhood of a core point.  
Noise or outlier: A point which is not a core point or border point.

* DBSCAN algorithm can be abstracted in the following steps –

1. Find all the neighbor points within eps and identify the core points or visited with more than MinPts neighbors.
2. For each core point if it is not already assigned to a cluster, create a new cluster.
3. Find recursively all its density connected points and assign them to the same cluster as the core point.  
   A point a and b are said to be density connected if there exist a point c which has a sufficient number of points in its neighbors and both the points a and b are within the eps distance. This is a chaining process. So, if b is neighbor of c, c is neighbor of d, d is neighbor of e, which in turn is neighbor of a implies that b is neighbor of a.
4. Iterate through the remaining unvisited points in the dataset. Those points that do not belong to any cluster are noise.

Below is the DBSCAN clustering algorithm in pseudocode:

DBSCAN(dataset, eps, MinPts){

# cluster index

C = 1

for each unvisited point p in dataset {

mark p as visited

# find neighbors

Neighbors N = find the neighboring points of p

if |N|>=MinPts:

N = N U N'

if p' is not a member of any cluster:

add p' to cluster C

}

1. Explain Hierarchical clustering (Agglomerative and Divisive clustering)

Ans: -

In [data mining](https://en.wikipedia.org/wiki/Data_mining) and [statistics](https://en.wikipedia.org/wiki/Statistics), hierarchical clustering (also called hierarchical cluster analysis or HCA) is a method of [cluster analysis](https://en.wikipedia.org/wiki/Cluster_analysis) which seeks to build a [hierarchy](https://en.wikipedia.org/wiki/Hierarchy) of clusters. Strategies for hierarchical clustering generally fall into two types:

**1. Agglomerative Clustering**:

Also known as bottom-up approach or hierarchical agglomerative clustering (HAC). A structure that is more informative than the unstructured set of clusters returned by flat clustering. This clustering algorithm does not require us to prespecify the number of clusters. Bottom-up algorithms treat each data as a singleton cluster at the outset and then successively agglomerates pairs of clusters until all clusters have been merged into a single cluster that contains all data.

Algorithm:

given a dataset (d1, d2, d3, ....dN) of size N

# compute the distance matrix

for i=1 to N:

# as the distance matrix is symmetric about

# the primary diagonal so we compute only lower

# part of the primary diagonal

for j=1 to i:

dis\_mat[i][j] = distance[di, dj]

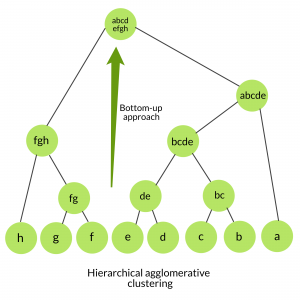
each data point is a singleton cluster

repeat

merge the two cluster having minimum distance

update the distance matrix

untill only a single cluster remains

  
   
**2. Divisive Clustering:**

Also known as top-down approach. This algorithm also does not require to prespecify the number of clusters. Top-down clustering requires a method for splitting a cluster that contains the whole data and proceeds by splitting clusters recursively until individual data have been spitted into singleton cluster.

Algorithm:

given a dataset (d1, d2, d3, ....dN) of size N

at the top we have all data in one cluster

the cluster is split using a flat clustering method eg. K-Means etc

repeat

choose the best cluster among all the clusters to split

split that cluster by the flat clustering algorithm

until each data is in its own singleton cluster

